Comprehensive Lightweight Application Security Process (CLASP)

Steven Lavenhar, Cigital, Inc. [vita[3]]

2005-11-30

The Comprehensive Lightweight Application Security Process (CLASP) is a set of formal practices that can help developers address security concerns throughout the software development life cycle. It is agnostic with respect to development methodology. A CLASP reference guide is available that includes templates, worksheets, checklists, and guidelines. The guide also describes a number of core concepts and principles for creating a secure application design.

Overview

The Comprehensive Lightweight Application Security Process (CLASP) is a set of formal practices that are designed to address security concerns in the early stages of the software development life cycle. Security problems can be introduced in every phase of the software development life cycle, and they are often the types of problems that developers are not well trained to address. CLASP addresses several key traditional software engineering activities, such as requirements specification. CLASP does not materially change the steps within SLDC steps. Instead, it recommends extensions to common artifacts and provides implementation guidance for security-specific content.

CLASP defines a set of 30 activities that are intended to improve security. CLASP is available both as a standalone process and in conjunction with the Rational Unified Process (RUP) environment. CLASP is agnostic with respect to development methodology and can be broadly applied. It is designed to allow development organizations to integrate activities as they see fit. CLASP is very concerned with prioritizing activities, determining the cost of activities, and helping developers understand the considerations in adopting an activity. The CLASP implementation guide provides that information on an activity-by-activity basis, and there are roadmaps to assist in selecting applicable activities. For example, the activity for detailing misuse cases provides information to help gain an understanding of the actors that are present in the system, identify defense mechanisms for misuse cases, and evaluate the misuse case results with stakeholders.

The roadmaps focus on common organizational requirements. There is a "legacy" roadmap for organizations that wish to implement elements of CLASP with a minimal impact on their existing developmental processes. There is also a "green field" roadmap for those organizations that are starting a new project and want to introduce CLASP as an essential part of their software development process.

CLASP provides tools such as templates, worksheets, checklists, and guidelines that cover the entire development process, from the development of the software's requirements through final testing and, if necessary, post-deployment incident response. CLASP's 30 activities are intended to integrate with current software development best practices. CLASP's activities include

- addressing reported security issues
- annotating class designs with security properties
- applying security principles to design

3. daisy:197 (Lavenhar, Steven)

- building an information labeling scheme
- building operational security guidelines
- designing user interfaces for security functionality
- detailing abuse and misuse cases
- documenting security design assumptions
- identifying global security policies
- identifying user roles and requirements
- identifying, implementing, and performing security tests
- implementing and elaborating resource policies
- implementing interface contracts
- instituting a security awareness program
- integrating security analysis into the build process
- managing System Security Authorization Agreements
- managing the certification process
- monitoring security metrics
- performing security analysis of system designs
- performing security functionality usability testing
- performing code signing
- performing security analysis of requirements
- performing software security fault injection testing
- performing source-level security review
- researching and assessing security solutions
- specifying database security configurations
- specifying resource-based security properties
- specifying the operational environment
- verifying security attributes of resources
- managing the security issue disclosure process

These 30 CLASP tasks are designed so that an organization can omit those activities that are not relevant to its development process. The developers of CLASP provide a 150-page reference guide that can be downloaded from http://www.securesoftware.com to help make the determination of which activities are appropriate given the specific requirements of the development organization.

In addition to the activities listed above, the developers of CLASP discuss a number of core concepts and principles on which a secure application design can be based. The fundamental security goals and core concepts discussed in CLASP include

- authorization and access control
- authentication
- confidentiality
- data integrity
- availability
- accountability
- non-repudiation
- insider threats as the weak link
- ethics in secure-software development
- input validation
- assuming the network is compromised
- minimizing the attack surface
- securing by default
- defense in depth
- principles for reducing exposure
- the Insecure Bootstrapping Principle

Cigital, Inc. Copyright

Fields

Name	Value
Copyright Holder	Cigital, Inc.

1. mailto:copyright@cigital.com

Fields

Name	Value
is-content-area-overview	false
Content Areas	Best Practices/Requirements Engineering
SDLC Relevance	Requirements
Workflow State	Publishable